(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 103 01 106 A 1**

(51) Int. Cl.[7]:
H 04 L 12/66
H 04 L 12/46

(21) Aktenzeichen: 103 01 106.4
(22) Anmeldetag: 9. 1. 2003
(43) Offenlegungstag: 7. 8. 2003

DE 103 01 106 A 1

(66) Innere Priorität:
102 03 697. 7 24. 01. 2002

(71) Anmelder:
Volkswagen AG, 38440 Wolfsburg, DE

(74) Vertreter:
Patentanwälte Effert, Bressel und Kollegen, 12489 Berlin

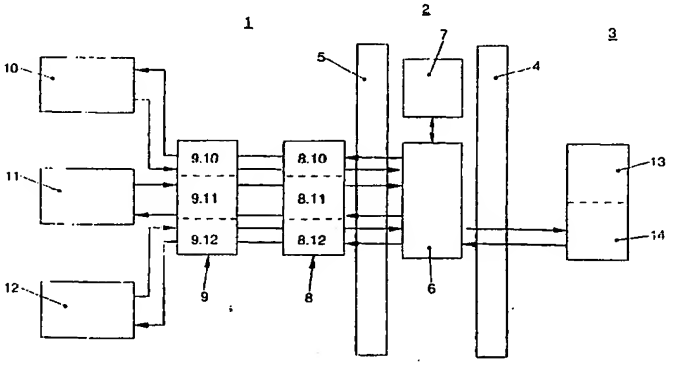
(72) Erfinder:
Strohmeyer, Jürgen, 38518 Gifhorn, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Rechercheantrag gem. Paragraph 43 Abs. 1 Satz PatG ist gestellt

(54) Verfahren und Vorrichtung zur Kommunikation zwischen Anwendungssystemen in unterschiedlichen Netzwerken

(57) Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Kommunikation zwischen Anwendungssystemen (101, 501) in unterschiedlichen Netzwerken (1, 5), wobei mindestens ein Anwendungssystem (101) in einem internen Netzwerk (1) und mindestens ein Anwendungssystem (501) in einem externen Netzwerk (5) eingebunden ist, mindestens dem internen Netzwerk eine Firewall zugeordnet ist, welche mindestens eine innere und äußere Firewall (201, 202) umfaßt, die Kommunikation zwischen den Anwendungssystemen (101-103, 501-503) jeweils als Client-Server-Interaktion über einen Verbindungsserver (20) erfolgt, wobei der Verbindungsserver (20) in einer demilitarisierten Zone DMZ (2) zwischen der inneren und äußeren Firewall (201, 202) angeordnet ist, die Netzwerke (1, 5) jeweils mit mindestens einem Client-Rechner (10, 50) ausgebildet sind und einem Client-Rechner (10, 50) mindestens ein Anwendungssystem (101-103, 501-503) zugeordnet ist.



DE 103 01 106 A 1

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Kommunikation zwischen Anwendungssystemen in unterschiedlichen Netzwerken.

[0002] Anwendungssysteme laufen häufig sowohl in einem internen Netzwerk als auch in externen Netzwerken, beispielsweise in Netzwerken verbundener Firmen ab. Sie tauschen Daten miteinander aus, verarbeiten diese und analysieren sie. Anwendungssystem in diesem Sinne kann beispielsweise ein Marktplatz im B2X Umfeld sein, welcher mit Basisdaten versorgt wird. Dabei gewinnt der weltweite Datenaustausch aufgrund der Globalisierung zunehmend an Bedeutung.

[0003] Der Austausch an Daten innerhalb eines Netzwerkes, beispielsweise innerhalb eines firmeninternen Intranets, kann bereits ohne größere Probleme realisiert werden. Die Sicherheit eines Intranets ist unter anderem abhängig vom Aufwand der in eine Sicherheitssoftware, beispielsweise eine Firewall, investiert wurde, welche das Intranet und Internet miteinander verbindet. Die Sicherheit ist weiter abhängig von der Strategie mit der die Sicherheitssoftware betrieben wird. Hat man hier die notwendige Sorgfalt walten lassen, ist das Intranet relativ sicher. Ein Datenaustausch zwischen Anwendungssystemen innerhalb eines Intranets ist daher ohne Verschlüsselung der Daten und ohne eine starke Authentifikation der Anwendungssysteme durchführbar.

[0004] Befinden sich die kommunizierenden Anwendungssysteme hingegen in unterschiedlichen Netzwerken, welche beispielsweise über das Internet Daten austauschen, ist eine starke Verschlüsselung der Daten und eine strenge Authentifikation der Anwendungssysteme erforderlich.

[0005] In die Geschäftsprozesse großer Unternehmen sind weltweit verteilte Firmen eingebunden. Sollen Anwendungssysteme, die in diesen Firmen betrieben werden, automatisch Daten miteinander austauschen, so muss der Aufwand dafür gering gehalten werden, um effizient und wirtschaftlich zu bleiben.

[0006] Der Erfindung liegt daher das technische Problem zugrunde, ein Verfahren und eine Vorrichtung zur verbesserten Kommunikation zwischen Anwendungssystemen in unterschiedlichen Netzwerken zu schaffen.

[0007] Die Lösung des technischen Problems ergibt sich durch die Gegenstände mit den Merkmalen der Patentansprüche 1, 17, 33 und 34. Weitere vorteilhafte Ausgestaltungen der Erfindung ergeben sich aus den Unteransprüchen.

[0008] Hierzu ist zwischen einem internen und einem externen Netzwerk eine Firewall aufgebaut, die mindestens eine dem internen Netzwerk zugeordnete äußere und eine innere Firewall umfasst, zwischen denen ein Verbindungsserver angeordnet ist und jedes Netzwerk mit mindestens einem Client-Rechner ausgebildet, wobei einem Client-Rechner mindestens ein Anwendungssystem zugeordnet ist. Die Kommunikation zwischen den Anwendungssystemen erfolgt jeweils als unabhängige Client-Server-Interaktion über den Verbindungsserver. Der Verbindungsserver unterstützt keine Dialogkommunikation zwischen Anwendungssystemen, sondern trennt die Kommunikation in zwei entkoppelte Übertragungsstrecken auf, so dass der tatsächliche Aufbau eines Anwendungssystems für das kommunizierende Anwendungssystem ohne Bedeutung ist. Für die Übertragung loggen sich die Anwendungssysteme mittels Client-Rechner jeweils auf den Verbindungsserver ein. Die Kommunikation verläuft asynchron, so dass ein gleichzeitiges Einloggen kommunizierender Anwendungssysteme nicht notwendig ist. Zwischen den Client-Rechnern und dem Verbindungsserver findet jeweils ein hochverschlüsselter Datentransfer statt, wohingegen der Datentransfer zwischen Anwendungssystemen eines gemeinsamen Netzwerks unverschlüsselt erfolgen kann.

[0009] In einer bevorzugten Ausführungsform ist einem Anwendungssystem auf dem Client-Rechner eine Client-Kommunikationskomponente zugeordnet, wobei die Client-Kommunikationskomponenten dynamisch aufgebaut werden. Die gesamte Kommunikationssoftware kann zentral vom Verbindungsserver zur Verfügung gestellt werden. Beispielsweise kann das Anwendungssystem auf einem Rechner mit Browser installiert sein, welcher eine JAVA Virtual Machine unterstützt. Die Kommunikationssoftware kann dann als Applet an den Rechner des Anwendungssystems übertragen und auf diesem automatisch installiert werden. Die Client-Kommunikationskomponenten generieren ein Interface, über das eine Kommunikation zwischen den Anwendungssystemen und den Client-Kommunikationskomponenten stattfindet.

[0010] In einer weiteren Ausführungsform werden auf dem Verbindungsserver zu den einzelnen aktiven Anwendungssystemen Server-Kommunikationskomponenten dynamisch aufgebaut.

[0011] In einer bevorzugten Ausführungsform umfasst eine Client-Kommunikationskomponente und/oder eine Server-Kommunikationskomponente mindestens ein Datei-Register und ein Control-Register. In das Datei-Register kann das Anwendungssystem beispielsweise zu übertragende Dateien einstellen. Die Control-Register dienen der Kommunikation und Steuerung zwischen den Anwendungssystemen und den Verbindungsservern und der Kommunikation und Steuerung der Anwendungssysteme untereinander. In der Verbindung Anwendungssystem – Client – Verbindungsserver findet hingegen keine Steuerung statt. Die Verbindungsserver werden alle dynamisch gleich strukturiert, d. h., die Statusschnittstellen der verschiedenen Verbindungsserver werden dynamisch gleich aktualisiert. Daneben ist es denkbar, in einem Status-Register Informationen über die Daten-Übertragung zu sammeln.

[0012] In einer bevorzugten Ausführungsform ist ein LDAP-Server vorgesehen, auf dem für die einzelnen Anwendungssysteme Identifier mit zugehörigen Passwords abgelegt sind. LDAP (Lightweight Directory Access Protocol) ist eine funktionell reduzierte Version des in iTU-T X.500 spezifizierten DAP für den einfachen Zugang zu Verzeichnisdiensten über TCP/iP-Netze.

[0013] In einer weiteren bevorzugten Ausführungsform ist auf dem LDAP-Server ein Policy Director installiert, in dem die Passwords gespeichert sind und eine Authentisierungsprüfung für jedes Anwendungssystem durchgeführt wird.

[0014] In einer weiteren bevorzugten Ausführungsform wird zur Kommunikation mit verschiedenen externen Netzwerken jedem externen Netzwerk jeweils ein eigener Verbindungsserver mit Firewall zugeordnet.

[0015] Vorzugsweise wird dabei ein zentraler LDAP-Server verwendet, der mit den verschiedenen Verbindungsservern verbunden ist, wobei auf dem LDAP-Server zu den Identifiern der Anwendungssysteme Attribute abgelegt sind, in denen das zugehörige externe Netzwerk eines Anwendungssystems abgelegt ist.

[0016] Zum automatischen Dateitransfer von einem Anwendungssystem an den Verbindungsserver (Upload) wird stellt das Anwendungssystem eine Datei zum Upload in seine Client-Kommunikationskomponente, vorzugsweise sein Upload-Datei-Register, und in Verbindung mit der Datei wird ein Upload-Request in die Client-Kommunikationskomponente, vorzugsweise das Upload-Control-Register geschrieben, wobei der Inhalt dieses Upload-Requests min-

destens eine Kennung und eine Identifikation des Anwendungssystems ist, zu dem die Datei übertragen werden soll. Die Kennung einer Datei kann beispielsweise ihre Bezeichnung, ihren Datei-Typ sowie das erstellende Anwendungssystem beihalten. Darüber hinaus sind zusätzliche Informationen wie Zeitpunkt der Erstellung, Größe der Datei etc. denkbar.

[0017] In einem weiteren Schritt wird mittels einer Upload-Funktion des Client-Rechners des Daten sendenden Anwendungssystems die Client-Kommunikationskomponente, vorzugsweise das entsprechende Control-Register gescanned und der Upload-Request gelesen, eine Verbindung zum Verbindungsserver aufgebaut und die Datei aus der Client-Kommunikationskomponente, vorzugsweise dem entsprechenden Datei-Register an den Verbindungsserver übertragen. Die Datei kann für die Übertragung in Unterdateien gesplittet oder durch ein geeignetes Verfahren komprimiert werden. Die Übertragung der Dateien erfolgt bevorzugt im https-Format.

[0018] Zum automatischen Dateitransfer vom Verbindungsserver an ein empfangendes Anwendungssystem (Download) wird in eine dem Anwendungssystem zugeordnete Server-Kommunikationskomponente des Verbindungsservers ein Download-Request gestellt. Über eine Downloadfunktion des Client-Rechners des empfangenden Anwendungssystems wird die Kommunikationskomponente gescanned und so der Download-Request erkannt.

[0019] Anschließend wird eine Verbindung von dem Anwendungssystem zum Verbindungsserver aufgebaut und die Datei an die dem Anwendungssystem zugeordnete Client-Kommunikationskomponente des empfangenden Client-Rechners übertragen, wobei mit der Datei ein Request in die Client-Kommunikationskomponente des Anwendungssystems geschrieben wird.

[0020] In einem weiteren Schritt scanned der Client-Rechner des empfangenden Anwendungssystems die Client-Kommunikationskomponente, liest den Request aus und aktiviert das Anwendungssystem.

[0021] In einer weiteren bevorzugten Ausführungsform wird bei einem Abbruch der Client-Server-Verbindung die Verbindung automatisch wieder aufgebaut, wobei vorzugsweise festgestellt wird, wieviel Übertragungen zum Zeitpunkt des Abbruchs innerhalb der Verbindung aktiv waren. Von den aktiven empfangenen Dateien wird der Bytecount an die Sender übermittelt, worauf dann die Sender die Counts der zu sendenden Dateien entsprechend einstellen und die Verbindung aktivieren.

[0022] In einer weiteren bevorzugten Ausführungsform wird nach dem Upload einer Datei auf dem Verbindungsserver eine Funktion aktiv, die feststellt, an welches oder an welche Anwendungssystem(e) die Datei übertragen werden soll. Für das/die entsprechende(n) Anwendungssystem(e) wird/werden dann Download-Requests in die zugehörigen Schnittstellen gestellt.

[0023] In einer bevorzugten Ausführungsform ist zu übertragenden Dateien ein Verfallsdatum zugeordnet. Die Dateien werden für einen Download auf dem Verbindungsserver zwischen gespeichert und nach Ablauf einer durch das Verfallsdatum gegebenen Frist werden die Dateien aus dem Speicher des Verbindungsservers entfernt. Vorzugsweise werden auch die dynamisch aufgebauten Kommunikationskomponenten mit dem Verfallsdatum auf dem Verbindungsserver und den Anwendungssystemen gelöscht.

[0024] Die Erfindung wird nachfolgend anhand eines bevorzugten Ausführungsbeispiels näher erläutert. Die einzige Figur zeigt:

[0025] **Fig.** 1 ein schematisches Blockschaltbild einer Vorrichtung zur Kommunikation zwischen Anwendungssystemen in verschiedenen Netzwerken.

[0026] Das Gesamtsystem umfasst im wesentlichen fünf Bereiche, nämlich ein internes Netzwerk (Intranet) **1**, ein externes Netzwerk **5**, eine dem internen Netzwerk **1** zugeordnete demilitarisierte Zone **2**, das Internet **3** und eine dem externen Netzwerk **5** zugeordnete demilitarisierte Zone **4**. Zwischen der demilitarisierten Zone **2** und dem Internet **3** ist eine äußere Firewall **202** und zwischen der demilitarisierten Zone **2** und dem Intranet **1** ist eine innere Firewall **201** aufgebaut. Zwischen der inneren Firewall **201** und der äußeren Firewall **202** sind ein Verbindungsserver **20** und ein LDAP-Server **21** angeordnet. Zur Abwicklung eines oder mehrerer Geschäftsprozesse werden interne Anwendungssysteme **101–103** sowie externe Anwendungssysteme **501–503** eingesetzt. Bei den Geschäftsprozessen handelt es sich beispielsweise um Geschäftsprozesse des B2X, aber auch jede andere Art von Prozessen ist denkbar. Die Anwendungssysteme **101–103** des Intranets **1** laufen beispielsweise auf einem gemeinsamen Rechner ab. Dieser Rechner dient auch gleichzeitig als Client-Rechner **10** für die Kommunikation mit dem Verbindungsserver **20**. Daneben ist es denkbar, dass die Anwendungssysteme **101–103** auf unterschiedlichen Rechnern ablaufen, wobei mindestens zwei Anwendungssysteme einem gemeinsamen Client-Rechner zugeordnet sein können und/oder der Rechner eines Anwendungssystems gleichzeitig Client-Rechner sein kann.

[0027] Auf dem Client-Rechner **10** ist ein nicht dargestellter Browser installiert, welcher beispielsweise eine JAVA Virtual Machine unterstützt. Für eine Kommunikation mit dem Verbindungsserver **20** sind auf dem Client-Rechner **10** spezielle Client-Kommunikationskomponenten **1011**, **1021**, **1031** installierbar. Für jedes Anwendungssystem **101**, **102**, **103**, welches dem Client-Rechner **10** zugeordnet ist, ist vorzugsweise eine eigene Client-Kommunikationskomponente **1011**, **1021**, **1031** vorhanden. In dem externen Netzwerk **5** ist mindestens ein externer Client-Rechner **50** angeordnet. Dem Client-Rechner **50** sind die Anwendungssysteme **501**, **502**, **503** zugeordnet. Dabei können die Anwendungssysteme **501–503** des externen Netzwerks **5** ebenfalls auf einem gemeinsamen Rechner und/oder getrennten Rechnern ablaufen. Das externe Netzwerk **5** kann außerdem ebenfalls als Intranet ausgebildet sein und ist vorzugsweise auch durch eine Sicherungssoftware, bestehend aus einer äußeren Firewall **402** und einer inneren Firewall **401** gegen unerlaubte Zugriffe aus dem Internet **3** geschützt. Weiter ist dem Client-Rechner **50** ein nicht dargestellter Browser zugeordnet und für jedes Anwendungssystem **501**, **502**, **503** ist eine Client-Kommunikationskomponente **5011**, **5021**, **5031** für eine Kommunikation aufbaubar. Vorzugsweise ist auf dem Verbindungsserver **20** für jedes Anwendungssystem **101–103**, **501–503** eine zugehörige Server-Kommunikationskomponente **2101–2103**, **2501–2503** aufgebaut.

[0028] Die Kommunikation zwischen dem Anwendungssystem **101** im Intranet **1** und dem Anwendungssystem **501** im externen Netzwerk **5** wird nun in zwei voneinander entkoppelte Client-Server-Interaktionen aufgespalten, nämlich zwischen einem dem Anwendungssystem **101** zugeordneten Client-Rechner **10** und dem Verbindungsserver **20** und einem dem Anwendungssystem **501** zugeordneten Client-Rechner **50** und dem Verbindungsserver **20**. Die Kommunikation erfolgt über die entsprechenden Client- und Server-Kommunikationskomponenten. Unter Client-Server-Interaktion wird dabei allgemein die Art und Weise der Zusammenarbeit gemäß vereinbarter Regeln zwischen einem Client-Rechner und einem Server zur Lösung einer Aufgabe in einem Client-Server-System verstanden. Zwischen den Client-Rechnern und den Verbindungsservern erfolgt ein hoch verschlüsselter Datentransfer, der vorzugsweise im

https-Format erfolgt. Der Datentransfer zwischen Client-Rechner und Anwendungssystem erfolgt üblicherweise unverschlüsselt.

[0029] Das System arbeitet vorzugsweise mit der signed Applet Technologie, dadurch kann die Client-Kommunikationskomponenten **1011, 1021, 1031, 5011, 5021, 5031** durch ein Applet dynamisch aufgebaut werden. Hierfür sendet das Anwendungssystem **101, 102, 103, 501, 502, 503** eine Login-Anfrage an den Verbindungsserver **20**. Der Verbindungsserver **20** überprüft die Zulässigkeit der Login-Anfrage eines Anwendungssystems **101–103, 501–503** anhand der Daten des LDAP-Servers **21**. Wird die Zulässigkeit der Login-Anfrage seitens des Verbindungsservers **20** bejaht, so sendet dieser ein Applet an den Browser des Client-Rechners **10, 50**. Das Applet wird in dem Browser gestartet und dadurch die entsprechende Client-Kommunikationskomponente **1011, 1021, 1031, 5011, 5021, 5031** aufgebaut. Eine Client-Kommunikationskomponente **1011, 1021, 1031, 5011, 5021, 5031** ist dabei mit mindestens einem Datei-Register und mindestens einem Control-Register erzeugbar. Bevorzugt werden für Upload und Download jeweils ein Datei-Register und ein Control-Register erzeugt. Nach einem Logout des Anwendungssystems **101–103, 501–503** kann die Client-Kommunikationskomponente **1011, 1021, 1031, 5011, 5021, 5031**, auf dem Client-Rechner **10, 50** wieder entfernt werden. Für einen dynamischen Aufbau einer Client-Kommunikationskomponente **1011, 1021, 1031, 5011, 5021, 5031** ist keine spezielle Kommunikationssoftware in Verbindung mit dem Anwendungssystem **101–103, 501–503** notwendig. Daher kann jedes autorisierte Anwendungssystem jederzeit in die Kommunikation eingebunden werden. Gleichzeitig sind verwendete Programme schnell und einfach aktualisierbar. Dabei müssen nur hinsichtlich der verwendeten Browser gewisse Abstimmungen getroffen werden. Für hochfrequentierte Anwendungssysteme **101–103** des Intranets **1** kann jedoch auch eine feste Installation zweckmäßig sein. Für Anwendungssysteme auf einem Rechner ohne Browser oder Java runtime Umgebung ist kein dynamischer Client-Aufbau durch ein Applet möglich. Derartige Anwendungssysteme können über einen NFS-Rechner mit einem Client-Rechner verbunden werden. Aus Sicherheitsaspekten ist ein dynamischer Aufbau der Client-Kommunikationskomponenten auf dem Client-Rechner des Anwendungssystems jedoch zu bevorzugen.

[0030] Prinzipiell kann alternativ oder kumulativ zu der signed Applet Technologie ein Programm mit Installations-Skript zur Verfügung gestellt werden, das vom Verbindungsserver **20** herunter geladen und installiert werden kann. Das Programm hat dabei die Funktionalität des Applets.

[0031] Der Verbindungsserver **20** überprüft die Identität des Client-Rechners **10, 50** und/oder des zugehörigen Anwendungssystems **101–103, 501–503**. Jedes interne Anwendungssystem **101–103** erhält vorzugsweise einen Identifier, der zusammen mit einem Password auf dem LDAP-Server **21** abgespeichert wird. Jedes externe Anwendungssystem **501–503** wird darüber hinaus vorzugsweise mit einem Attribut abgespeichert. Dieses Attribut beinhaltet eine Information, über welchen Verbindungsserver **20** und/oder über welches externe Netzwerk **5** ein externes Anwendungssystem **501–503** von einem Anwendungssystem **101–103** im Intranet **1** erreichbar ist. Jeder Verbindungsserver **20** lädt sich die Gesamtheit aller im LDAP-Server **21** gespeicherten Identifier herunter und erstellt für diese Server-Kommunikationskomponenten. Diese beiden Funktionen werden für neu definierte LDAP-Einträge auf Anforderung oder zyklisch wiederholt. Des weiteren kann auf dem LDAP-Server **21** ein Policy Director installiert werden, der überprüft, ob die geplanten Kommunikationen bzw. Zugriffe den Zugriffsrechten des jeweiligen Anwendungssystems entsprechen. Gemäß der dargestellten Ausführungsform ist der zentrale LDAP-Server **21** zwischen den Firewalls **201, 202** angeordnet. Daneben ist es auch denkbar, den LDAP-Server **21** im Intranet **1** anzuordnen.

[0032] Der Datentransfer läßt sich dabei in folgende Schritte unterteilen:

Das Anwendungssystem **101**, welches eine Datei an das Anwendungssystem **501** übertragen möchte, stellt die Datei zum Upload für den Verbindungsserver **20** in die Client-Kommunikationskomponente **1011**, vorzugsweise in ein Upload-Datei-Register. In Verbindung mit dieser Datei wird ein Request in die Client- Kommunikationskomponente **1011** geschrieben, vorzugsweise in ein Upload-Control-Register. Dem Request ist mindestens das Ziel-Anwendungssystem **501** und eine Kennung der zu übertragenden Datei zu entnehmen.

[0033] Eine Upload-Funktion des Client-Rechners **10** scanned in periodischen Abständen die zugehörige Client-Kommunikationskomponente **1011**. Der Request wird eingelesen und der Client-Rechner **10** baut eine Verbindung zum Verbindungsserver **20** auf. Der Verbindungsserver **20** ist mit einer Server-Kommunikationskomponente **2101** ausgebildet, in welche der Client-Rechner **10** des Anwendungssystems **101** sein Request einstellen kann. Auf dem Verbindungsserver **20** wird entsprechend des Requests eine Speicherstelle eingerichtet, in welche der Client-Rechner **10** die zu übertragende Datei einstellen kann und der Client-Rechner **10** überträgt die Datei zum Verbindungsserver **20**.

[0034] Entsprechend des Requests wird auf dem Verbindungsserver **20** eine Schnittstelle **2501**, welche dem Anwendungssystem **501** zugeordnet ist, aktualisiert. Eine Downloadfunktion des Client-Rechners **50** des empfangenden Anwendungssystems **501** scanned die Schnittstelle **2501**, erkennt den Request, baut eine Verbindung zum Verbindungsserver **20** auf und lädt die Datei in die Client-Kommunikationskomponente **5011** des Client-Rechners **50** herunter, vorzugsweise in ein entsprechendes Download-Datei-Register. In Verbindung mit dem Speichern der Datei wird der Request in die Client-Kommunikationskomponente **5011** geschrieben, vorzugsweise in ein entsprechendes Download-Control-Register. Dem Request kann das Anwendungssystem **501** beispielsweise das sendende Anwendungssystem **101** entnehmen. Es ist auch denkbar, einen Verwendungszweck durch den Request zu übermitteln. Gemäß der Envelope-Technik erweitern der Client-Rechner **10** des sendenden und/oder des empfangenden Anwendungssystems und/oder der Verbindungsserver **20** den Request des Anwendungssystems **101** entsprechend ihrer Bedürfnisse.

[0035] Erfolgt ein Abbruch der Client-Server-Verbindung, so wird die Verbindung automatisch neu gestartet. Es wird festgestellt, wieviel Übertragungen zum Zeitpunkt des Abbruchs innerhalb der Verbindung aktiv waren. Von den aktiven empfangenen Dateien wird der Bytecount an den oder die Sender, d. h. entweder den Client-Rechner für einen Upload oder den Verbindungsserver bei einem Download, übermittelt. Die Sender stellen die zu sendenden Dateien auf diese Counts ein und aktivieren die Verbindung.

[0036] Des weiteren ist es auch möglich, dass eine Datei an mehrere Anwendungssysteme verteilt werden soll. Nach dem Upload einer Datei wird auf dem Verbindungsserver **20** daher eine Funktion aktiv, die feststellt, an welche Anwendungssysteme die Datei verteilt werden soll. Für jedes dieser Anwendungssysteme wird ein entsprechender Download-Request in eine ihm zugeordnete Kommunikationskomponente **2101–2103, 2501–2503** auf dem Verbindungsserver **20** gestellt.

[0037] Es kann sein, dass die Anwendungssysteme in den externen Netzwerken über unterschiedliche Verbindungsserver erreichbar sind.

[0038] Die Client-Kommunikationskomponente **1011–1103, 5011–5031** umfasst beispielsweise ein Datei-Register, ein Control-Register und ein Status-Register. In das Datei-Register werden Dateien für eine Übertragung und/oder empfangenen Dateien abgelegt. Das Datei-Register der Client-Kommunikationskomponente **1011–1031, 5011–5031** ist mindestens durch das zugehörige Anwendungssystem **101–103, 501–503** zugänglich. Teilweise kann weiter ein Zugriff durch andere Programme des zugehörigen Client-Rechners **10, 50** sinnvoll sein. Das Datei-Register ist gegen einen unberechtigten Zugriff geschützt. Ein Request für einen Upload oder einen Download einer im Datei-Register abgelegten Datei wird in das Control-Register geschrieben. Informationen über den Verlauf der Übertragung, beispielsweise eine Antwort des Verbindungsservers auf erfolgreichen Upload oder Fehlermeldungen werden in das Status-Register geschrieben. Je nach Detaillierungsgrad eines Requests ist ein Schutz des Control-Registers gegen unberechtigten Zugriff notwendig. Das Status-Register unterliegt in der Regel keinen Sicherheitsvorkehrungen.

[0039] Ein Anwendungssystem **101** im Intranet **1** prüft vor der Initiierung eines Upload-Requests, d. h. vor einer Übertragung einer Datei an den Verbindungsserver **20**, ob die Identifier aller empfangenden, externen Anwendungssysteme **501–503** in einer Distribution-List auf dem LDAP-Server definiert worden sind. Ist das nicht der Fall, entscheidet das Anwendungssystem **101**,

- ob der Upload-Request trotzdem initiiert wird und die Datei so weit wie möglich an externe Anwendungssysteme verteilt wird oder
- ob die Verteilung erst nach der Nachdefinition der fehlenden Identifier im LDAP-Server/Policy Director durchgeführt wird.

[0040] In beiden Fällen stellt der Verbindungsserver automatisch fest, dass die Identifier nachdefiniert wurden und schließt die Verteilung ab.

[0041] Die Übertragungsstrecke zwischen zwei Anwendungssystemen ist transparent – sowohl für den Beobachter auf der Sendeseite als auch für den Beobachter auf der Empfangsseite. Jedes Ereignis auf dieser Strecke (Bereitstellung des Files, Start und Ende Komprimierung, Start und Ende der Session, ...) wird mit einem Timestamp versehen. Diese Informationen werden in Status-Registern der Kommunikations-Schnittstellen der Anwendungssysteme gespeichert. Daneben werden sie auch in den speziellen Anwendungssystemen zugeordneten Status-Schnittstellen auf dem Verbindungsserver **20** abgelegt. Auf die Status-Schnittstellen kann zugegriffen werden, solange eine Transaktionsbeziehung zwischen dem Verbindungsserver und dem Anwendungssystem besteht. Das Setzen bzw. Auslesen der Statusschnittstellen kann unverschlüsselt über XML/HTML erfolgen. Die Daten der Status-Schnittstellen können grafisch aufbereitet über HTML-Seiten anwendungssystemspezifisch abgerufen werden. Dazu muß man sich über einen separaten Account in den Verbindungsserver **20** einloggen.

[0042] Die Übertragung von Daten vom Anwendungssystem **501** zum Anwendungssystem **101** erfolgt entsprechend umgekehrt.

Patentansprüche

1. Vorrichtung zur Kommunikation zwischen Anwendungssystemen in unterschiedlichen Netzwerken, wobei mindestens ein Anwendungssystem in einem internen Netzwerk und mindestens ein Anwendungssystem in einem externen Netzwerk eingebunden ist, mindestens dem internen Netzwerk eine Firewall zugeordnet ist und die Firewall mindestens eine innere und äußere Firewall umfasst, **dadurch gekennzeichnet**, dass
in einer demilitarisierten Zone DMZ (2) zwischen der inneren und äußeren Firewall (**201, 202**) ein Verbindungsserver (**20**) angeordnet ist und die Netzwerken (**1, 5**) mit jeweils mindestens einem Client-Rechner (**10, 50**) ausgebildet sind,
einem Client-Rechner (**10, 50**) mindestens ein Anwendungssystem (**101–103, 501–503**) zugeordnet ist, und
die Kommunikation zwischen den Anwendungssystemen (**101–103, 501–503**) jeweils als Client-Server-Interaktion über den Verbindungsserver (**20**) erfolgt.

2. Vorrichtung nach Anspruch 1, dadurch gekennzeichnet, dass mindestens einem Anwendungssystem (**101–103, 501–503**) auf dem Client-Rechner (**10, 50**) eine Client-Kommunikationskomponente (**1011–1031, 5011–5031**) zugeordnet ist, wobei die Client-Kommunikationskomponente (**1011–1031, 5011–5031**) dynamisch aufbaubar ist.

3. Vorrichtung nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass auf dem Verbindungsserver (**20**) zu den einzelnen Anwendungssystemen (**101–103, 501–503**) Server-Kommunikationskomponenten (**2101–2103, 2501–2503**) erstellbar sind.

4. Vorrichtung nach Anspruch 2 oder 3, dadurch gekennzeichnet, dass eine Client-Kommunikationskomponente (**1011–1031, 5011–5031**) und/oder eine Server-Kommunikationskomponente (**2101–2103, 2501–2503**) mindestens ein Datei-Register und ein Control-Register umfasst.

5. Vorrichtung nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass die Vorrichtung einen LDAP-Server (**21**) umfaßt, auf dem für die einzelnen Anwendungssysteme (**101–103, 501–503**) Identifier mit zugehörigen Passwords abgelegt sind.

6. Vorrichtung nach Anspruch 5, dadurch gekennzeichnet, dass auf dem LDAP-Server (**21**) ein Policy Director installiert ist.

7. Vorrichtung nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass verschiedenen externen Netzwerken (**5**) jeweils ein eigener Verbindungsserver (**20**) mit Firewall (**201, 202**) zugeordnet ist.

8. Vorrichtung nach Anspruch 7, dadurch gekennzeichnet, dass die verschiedenen Verbindungsserver (**20**) mit dem zentralen LDAP-Server (**21**) verbunden sind, wobei bei auf dem LDAP-Server (**21**) zu den Identifiern Attribute abgelegt sind, in denen das zugehörige Netzwerk (**5**) eines Anwendungssystems (**501–503**) abgelegt ist.

9. Vorrichtung nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass zum automatischen Dateitransfer von einem sendenden Anwendungssystem (**101**) an den Verbindungsserver (**20**) eine Datei zum Upload in die Client-Kommunikationskomponente (**1011**) stellbar ist, in Verbindung mit dieser Datei ein Upload-Request in die Client-Kommunikationskomponente (**1011**) schreibbar ist, wobei der Inhalt des Upload-Requests mindestens eine Kennung der zu übertragenden Datei und eine Identifikation des Anwendungssystems (**501**) ist, zu dem die Datei übertragen werden soll.

10. Vorrichtung nach Anspruch 9, dadurch gekennzeichnet, dass mittels einer Upload-Funktion des

Client-Rechners des sendenden Anwendungssystems (**101**) die Client-Kommunikationskomponente (**1011**) scannbar und der Upload-Request lesbar ist, eine Verbindung zum Verbindungsserver (**20**) aufbaubar und die Datei an den Verbindungsserver (**20**) übertragbar ist.

11. Vorrichtung nach Anspruch 9 oder 10, dadurch gekennzeichnet, dass zum automatischen Dateitransfer vom Verbindungsserver (**20**) an ein empfangendes Anwendungssystem (**501**) eine dem Anwendungssystem (**501**) zugeordnete Server-Kommunikationskomponente (**2501**) mit einem Download-Request aktualisierbar ist, über eine Downloadfunktion des Client-Rechners des empfangenden Anwendungssystems (**501**) die Server-Kommunikationskomponente (**2501**) scannbar und der Download-Request erkennbar ist, eine Verbindung zum Verbindungsserver (**20**) aufbaubar und die Datei an die Client-Kommunikationskomponente (**5011**) des dem Anwendungssystems (**501**) zugeordneten Client-Rechners (**50**) übertragbar ist und ein Request in die Client-Kommunikationskomponente (**5011**) schreibbar ist.

12. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, dass der Client-Rechner (**50**) des empfangenden Anwendungssystems (**501**) die Client-Kommunikationskomponente (**5011**) scanned, den Request ausliest und das Anwendungssystem (**501**) aktiviert.

13. Vorrichtung nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass bei Abbruch einer Verbindung zwischen Client-Rechner (**10, 50**) und Verbindungsserver (**20**) ein automatischer Neuaufbau der Verbindung herstellbar ist.

14. Vorrichtung nach Anspruch 13, dadurch gekennzeichnet, dass die Anzahl der beim Abbruch aktiven Übertragungen ermittelbar ist und von den aktiven empfangenen Dateien ein Bytecount an die Sender übertragbar ist, wobei durch die Sender die Counts der zu sendenden Dateien auf das übertragene Bytecount einstellbar und die Übertragung aktivierbar ist.

15. Vorrichtungen nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass durch ein Upload einer Datei auf dem Verbindungsserver (**20**) eine Funktion aktivierbar ist, mittels derer ermittelbar ist, an welche Anwendungssysteme die Datei verteilt werden soll.

16. Vorrichtungen nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass die zu übertragenden Dateien auf dem Verbindungsserver (**20**) speicherbar sind, wobei den Dateien ein Verfallsdatum zugeordnet ist.

17. Verfahren zur Kommunikation zwischen Anwendungssystemen in unterschiedlichen Netzwerken, wobei mindestens ein Anwendungssystem in einem internen Netzwerk und mindestens ein Anwendungssystem in einem externen Netzwerk eingebunden ist, mindestens dem internen Netzwerk eine Firewall zugeordnet ist und die Firewall mindestens eine innere und äußere Firewall umfasst, dadurch gekennzeichnet, dass die Kommunikation zwischen den Anwendungssystemen (**101–103, 501–503**) jeweils als Client-Server-Interaktion über einen Verbindungsserver (**20**) erfolgt, wobei der Verbindungsserver (**20**) in einer demilitarisierten Zone DMZ (**2**) zwischen der inneren und äußeren Firewall (**201, 202**) angeordnet ist, die Netzwerke (**1, 5**) jeweils mit mindestens einem Client-Rechner (**10, 50**) ausgebildet sind und einem Client-Rechner (**10, 50**) mindestens ein Anwendungssystem (**101–103, 501–503**) zugeordnet ist.

18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, dass mindestens eine Client-Kommunikationskomponente (**1011–1031, 5011–5031**) dynamisch aufgebaut wird, wobei die Client-Kommunikationskomponente (**1011–1031, 5011–5031**) mindestens einem Anwendungssystem (**101–103, 501–503**) auf dem Client-Rechner (**10, 50**) zugeordnet ist.

19. Verfahren nach Anspruch 17 oder 18 dadurch gekennzeichnet, dass auf dem Verbindungsserver (**20**) zu den einzelnen Anwendungssystemen (**101–103, 501–503**) Server-Kommunikationskomponenten (**2101–2103, 2501–2503**) erstellt werden.

20. Verfahren nach Anspruch 18 oder 19, dadurch gekennzeichnet, dass eine Client-Kommunikationskomponente (**1011–1031, 5011–5031**) und/oder eine Server-Kommunikationskomponente (**2101–2103, 2501–2503**) mindestens ein Datei-Register und ein Control-Register umfasst.

21. Verfahren nach einem der Ansprüche 17 bis 20 mittels eines LDAP-Servers, dadurch gekennzeichnet, dass auf dem LDAP-Server (**21**) für die einzelnen Anwendungssysteme (**101–103, 501–503**) Identifier mit zugehörigen Passwords abgelegt sind.

22. Verfahren nach Anspruch 21, dadurch gekennzeichnet, dass auf dem LDAP-Server (**21**) ein Policy/Director installiert ist, in dem die Passwords abgefegt sind und die Zugangsberechtigung eines Anwendungssystems (**101–103, 501–503**) überprüft wird.

23. Verfahren nach einem der Ansprüche 17 bis 22, dadurch gekennzeichnet, dass verschiedenen externen Netzwerken (**5**) jeweils ein eigener Verbindungsserver (**20**) mit Firewall (**201, 202**) zugeordnet ist.

24. Verfahren nach Anspruch 23, dadurch gekennzeichnet, dass die verschiedenen Verbindungsserver (**20**) mit dem zentralen LDAP-Server (**21**) verbunden sind, wobei auf dem LDAP-Server (**21**) zu den Identifiern Attribute abgelegt sind, in denen das zugehörige Netzwerk (**5**) eines Anwendungssystems (**501–503**) abgelegt ist.

25. Verfahren nach einem der Ansprüche 17 bis 24, dadurch gekennzeichnet, dass zum automatischen Dateitransfer von einem sendenden Anwendungssystem (**101**) an den Verbindungsserver (**20**) eine Datei zum Upload in eine Client-Kommunikationskomponente (**1011**) des dem sendenden Anwendungssystem (**101**) zugeordneten Client-Rechners (**10**) gestellt wird, in Verbindung mit dieser Datei ein Upload-Request in die Client-Kommunikationskomponente (**1011**) geschrieben wird, wobei der Inhalt des Upload-Requests mindestens ein Kennung der zu sendenden Datei und eine Identifikation des Anwendungssystems ist, zu dem die Datei übertragen werden soll.

26. Verfahren nach Anspruch 25, dadurch gekennzeichnet, dass mittels einer Upload-Funktion des Client-Rechners (**10**) vom Daten sendenden Anwendungssystem (**101**) die Client-Kommunikationskomponente (**1011**) gescanned wird, der Upload-Request gelesen, eine Verbindung zum Verbindungsserver (**20**) aufgebaut und die Datei an den Verbindungsserver (**20**) übertragen wird.

27. Verfahren nach Anspruch 25 oder 26, dadurch gekennzeichnet, dass zum automatischen Dateitransfer vom Verbindungsserver (**20**) an ein empfangendes Anwendungssystem (**501**) eine dem Anwendungssystem (**501**) zugeordnete Server-Kommunikationskomponente (**2501**) durch ein Download-Request aktualisiert wird, über die Downloadfunktion des Client-Rechners (**50**) des empfangenden Anwendungssystems (**501**) die

Server-Kommunikationskomponente (**2501**)-gescanned wird, der Download-Request erkannt wird, eine Verbindung zum Verbindungsserver (**20**) aufgebaut wird und die Datei und ein Request in eine Client-Kommunikationskomponente (**5011**) des Client-Rechners (**50**) geschrieben werden.

28. Verfahren nach Anspruch 27, dadurch gekennzeichnet, dass der Client-Rechner (**50**) des empfangenden Anwendungssystems (**501**) die Client-Kommunikationskomponente (**5011**) scanned, den Request ausliest und das Anwendungssystem (**501**) aktiviert.

29. Verfahren nach einem der Ansprüche 17 bis 28, dadurch gekennzeichnet, dass bei Abbruch einer Verbindung zwischen Client-Rechner (**10, 50**) und Verbindungsserver (**20**) ein automatischer Neuaufbau der Verbindung hergestellt wird.

30. Verfahren nach Anspruch 29, dadurch gekennzeichnet, dass die Anzahl der beim Abbruch aktiven Übertragungen ermittelt wird, von den aktiven empfangenen Dateien ein Bytecount an die Sender übertragen wird, wobei die Sender die Counts der zu sendenden Dateien auf das übertragene Bytecount einstellen und die Übertragung aktivieren.

31. Verfahren nach einem der Ansprüche 17 bis 30, dadurch gekennzeichnet, dass durch ein Upload einer Datei auf dem Verbindungsserver (**20**) eine Funktion aktiviert wird, mittels derer ermittelt wird, an welche Anwendungssysteme die Datei verteilt werden soll.

32. Verfahren nach einem der Ansprüche 17 bis 31, dadurch gekennzeichnet, dass die zu übertragenden Dateien auf dem Verbindungsserver (**20**) gespeichert werden, wobei den Dateien ein Verfallsdatum zugeordnet ist.

33. Computerprogramm mit Programmcode-Mitteln, um alle Schritte von jedem beliebigen der Ansprüche 17 bis 32 durchzuführen, wenn das Programm auf einem Computer ausgeführt wird.

34. Compunterprogrammprodukt mit Programmcode-Mitteln, die auf einem computerlesbaren Datenträger gespeichert sind, um das Verfahren nach jedem beliebigen der Ansprüche 17 bis 32 durchzuführen, wenn das Programmprodukt auf einem Computer ausgeführt wird.

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

Nummer: DE 103 01 106 A1
Int. Cl.[7]: H 04 L 12/66
Offenlegungstag: 7. August 2003

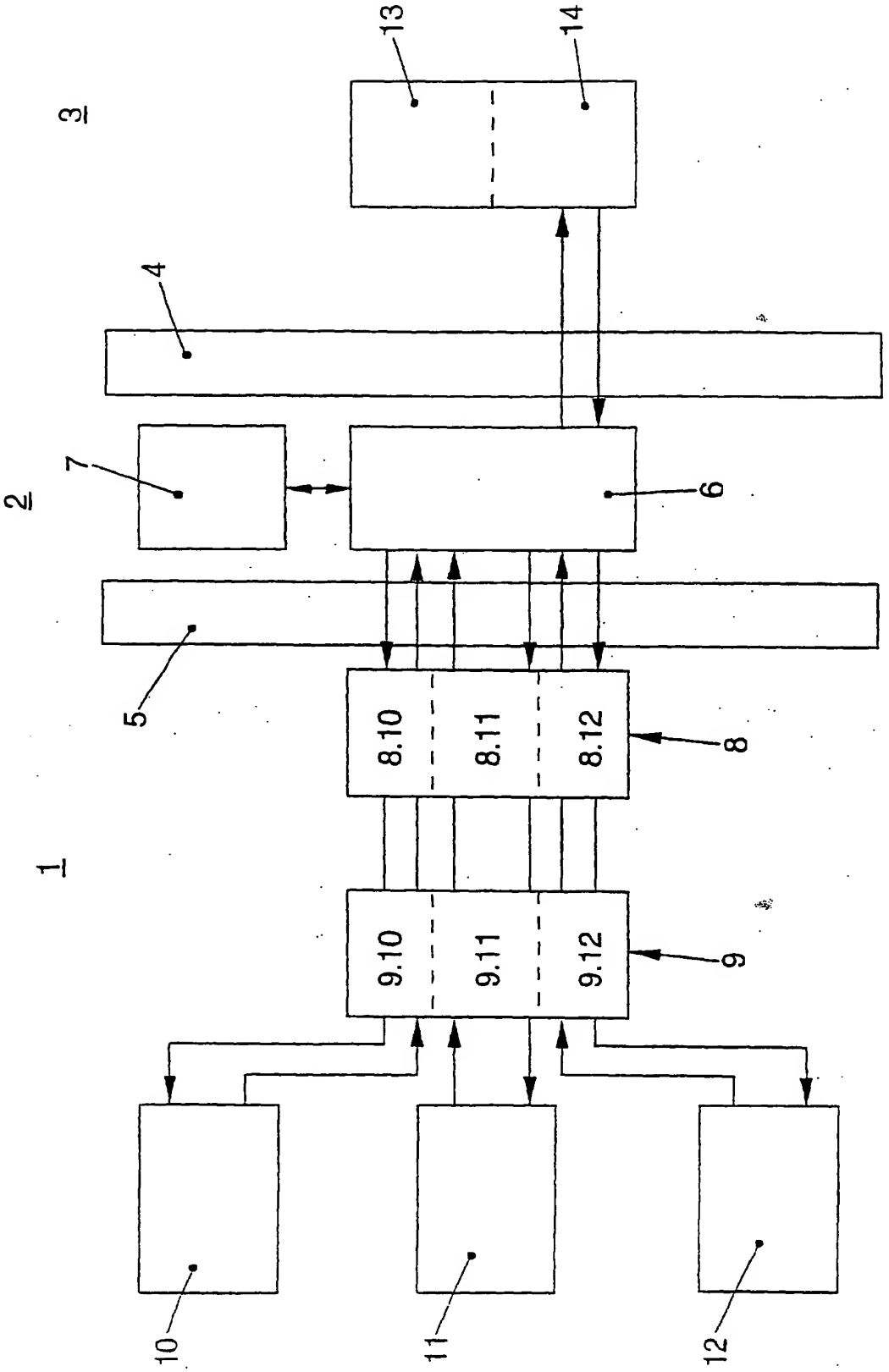


FIG. 1

Nummer: DE 103 01 106 A1
Int. Cl.[7]: H 04 L 12/66
Offenlegungstag: 7. August 2003

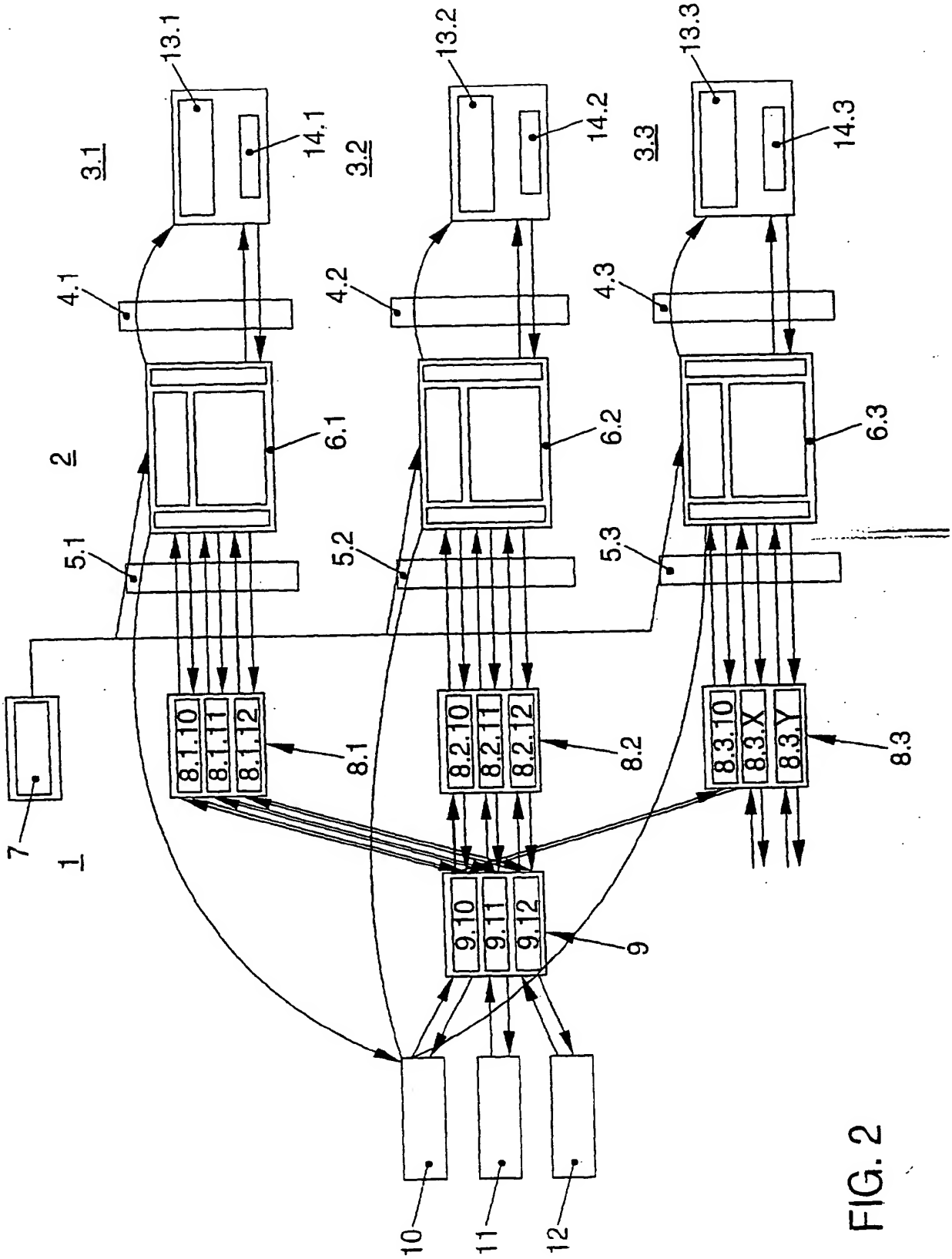


FIG. 2